Design Review 1

ECE4332 Introduction to VLSI Design

Jacob Breiholz
Leiqing Cai
Ashley Morse
Qing Qin

10/7/2014

Introduction

The overarching goal of the project is to design an embedded SRAM to win a contract from PICo. We have decided that high speed will be the core feature of our memory design. In order to convince PICo that we have the best SRAM design, we present Design Review 1 which comprises of diagrams and simulations of a functioning SRAM with one block of memory unit.

Key Parameters

Table 1 shows the key parameters of the memory that we decided to implement. Because the purpose of the design review is to demonstrate the functionality, we only implemented one block of the memory. Therefore, the total size of the memory we implemented was 4 kbit.

Table 2 shows the inputs/outputs of the cache. In this review, inputs expected for the clock is assumed to be outputs of a register that change 1FO4 after the rising clock edge. Also, READ and WRITE signal are assumed never to be high at the same time.

Table 1 Parameters of the High Speed Cache

Parameter	Value
Total Memory Size	64 kbit
Word Size	32 bit
# of Blocks	16
# of Columns/Block	64
# of Rows/Block	64
# of Words/Row	2

Table 2 Inputs and Outputs of the High Speed Cache

Direction	Pins	Description
Input	CLK	Clock Signal
Input	IN<31:0>	Data Inputs for Write Operation
Input	READ	Read Control Signal
Input	WRITE	Write Control Signal
Input	ADDR<10:0>	Word Address
Output	OUT<31:0>	Data Outputs for Read Operation

We chose our memory to be word-addressable. Since the total memory size is 2^{16} bit while each word has 2^5 bit, there is a total of 2^{11} words. Therefore, we need 11 bits in order to identify each word.

We have partitions the address to tell us exactly where the word we desire is. ADDR<3:0> feeds into the block decoder and tells us which block contains the word. ADDR<10:5> feeds into the row decoder to pick the row. ADDR<4> tells us which of the 2 columns in the block has the word.

Table 3 summaries the breakdown of the address space. Note that in this design review, the 4-bit block address is unused, since only one block was implemented.

Table 3 Partition of the Address Space

Pins	Description
ADDR<10:5>	Row Address (64 rows/block)
ADDR<4>	Column Address (2 words/row)
ADDR<3:0>	Block Address (16 blocks)

Schematics

Figure 1 shows the block diagram of the SRAM. Intersections with wires going in four directions are considered **not** connected.

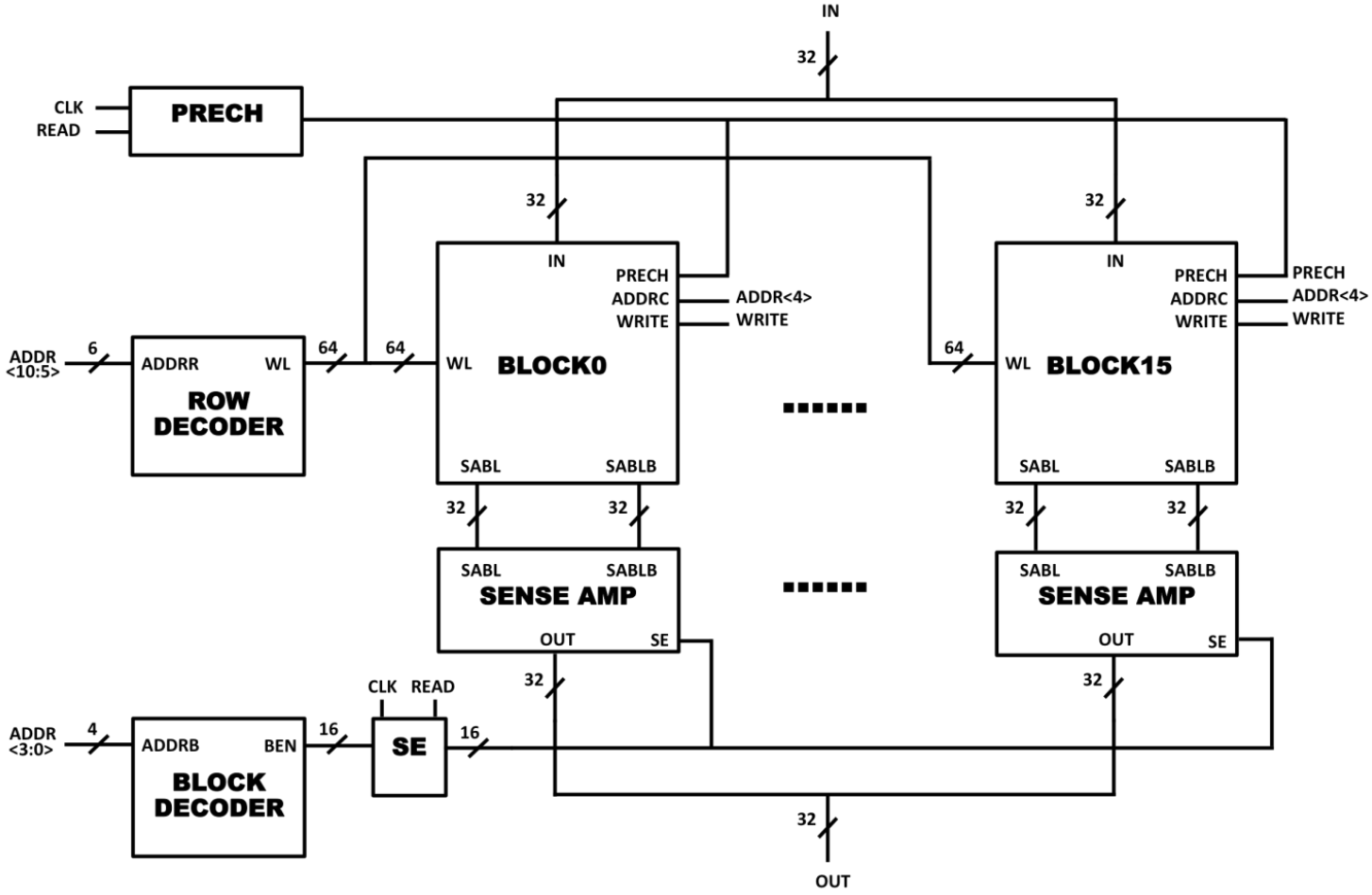

Figure 1 Block Diagram of the High Speed Cache

Figure 2 goes in more detail by showing the gate level diagram of one block.

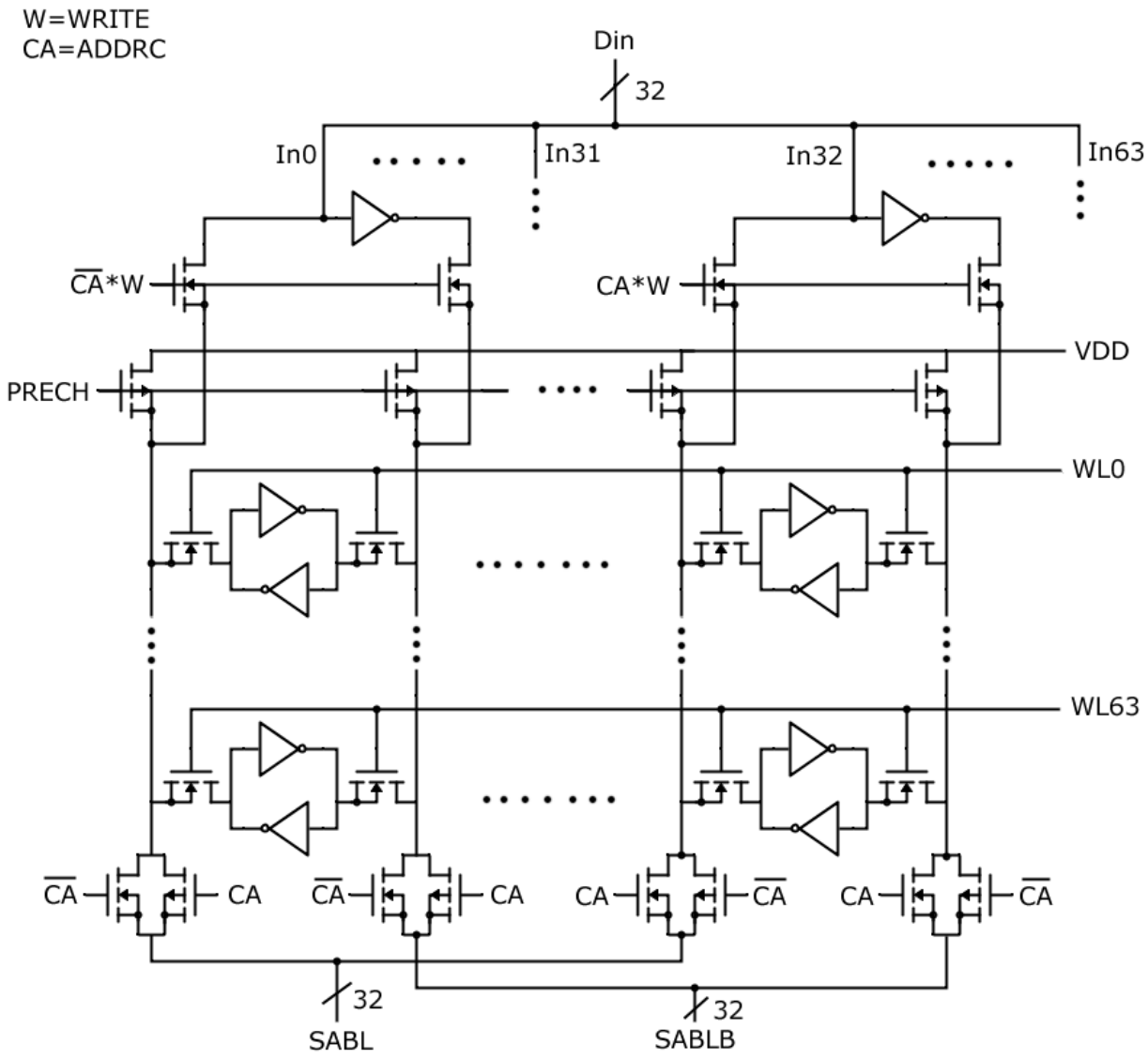

Figure 2 Gate-Level Schematic of Block

Layout

Figure 3 shows the layout for a SRAM bit cell. The layout can also be accessed with the link below:

http://venividiwiki.ee.virginia.edu/mediawiki/index.php/File:SRAM_Layout.png

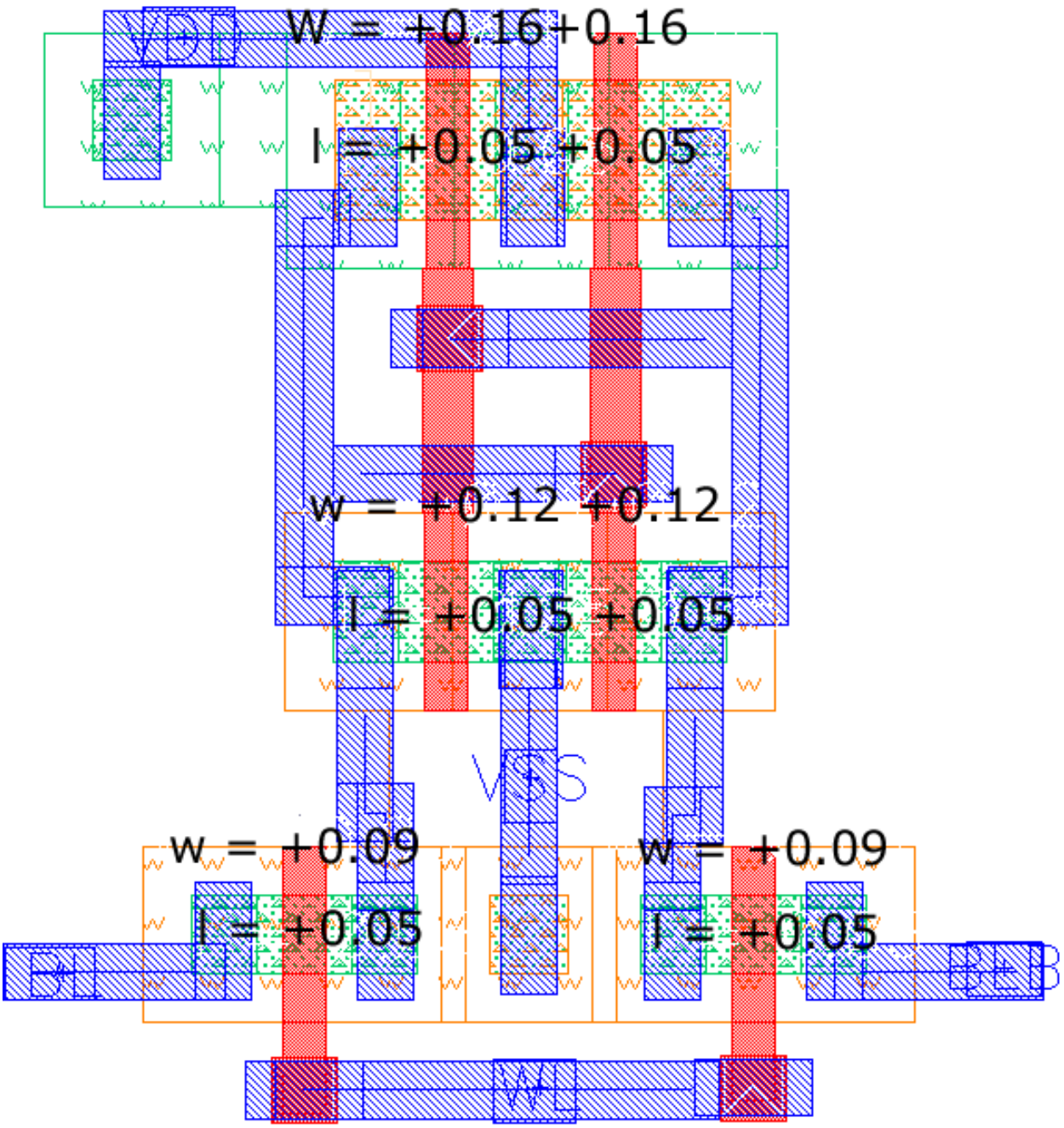

Figure 3 Layout of SRAM Bit Cell

Timing Diagram

Figure 4 shows the timing diagram of the read and write operations of the cache. PRECH control the pre-charge of BL and BLB, and is on (active low) on the first half cycle of a read operation. SE stands for Sense-Amplifier Enable, and is high only on the second half of a read cycle.

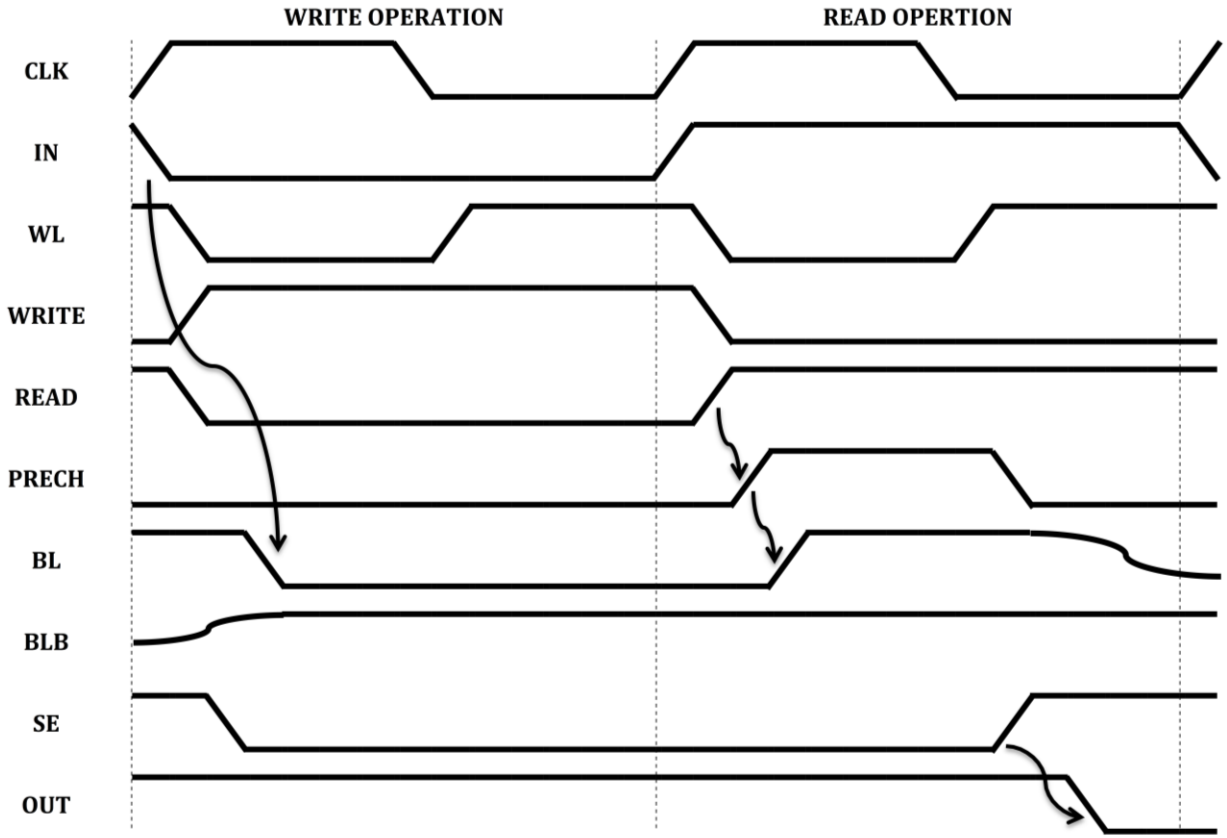

Figure 4 Timing Diagram of Write and Read Operations

Design Verification

The cache design was verified using a randomly chosen test case as shown in Table 4. The cache wrote the input data into the first 32 bit cells on Word Line 42 during the first clock period, when the signal WRITE is asserted. The stored data was then read during the second clock period, when the signal READ is asserted. Simulation results on page 9 to page 11 show a Typical-Typical case and results on page 12 to page 13 show the same read operation at a Fast-Fast corner.

Table 4 Test Case for Cache Design Verification

Pins	Values
IN <31:0>	$1101_{28}\ 1111_{24}\ 0110_{20}\ 0101_{16}\ 1010_{12}\ 1111_{8}\ 0111_{4}\ 0101_{0}$
ADDR <10:4>	1010100

Figure 5 presents the simulation results on the control signals of the cache, including CLK, WL42, READ, WRITE, PRECH and SE. PRECH goes low when write when reading occurs, charging both BL and BLB, and goes high in the second half of the read cycle, causing one of the bit lines to discharge. SE goes high to activate the Sense Amp, thus detecting what has been read from the bitcell.

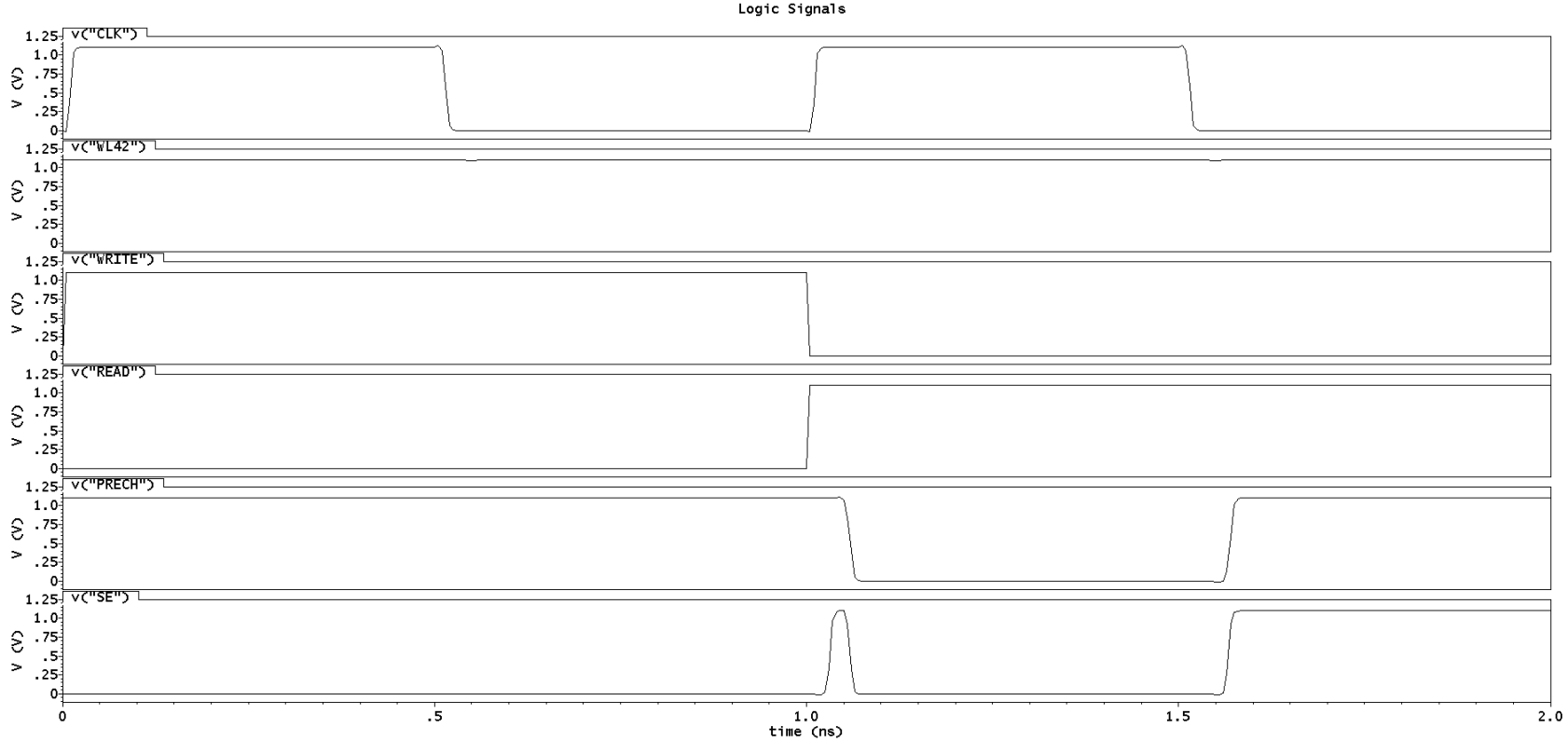

Figure 5 Simulation Results for Control Signals

Figure 6 shows the local wordline LWL42, BL/BLB10 and BL/BLB12 of the cache during Typical-Typical write and read operations. Bit Line 10 and 12 have been randomly selected as primary signals to demonstrate the write and read operations of the cache. Bit Line 10 is written in a 1 and Bit Line 12 is written in a 0 during the first clock period. The values are flipped here during read operation, which creates a problem for the Typical-Typical simulation. There is something wrong with our write operation, because the values are not stored properly.

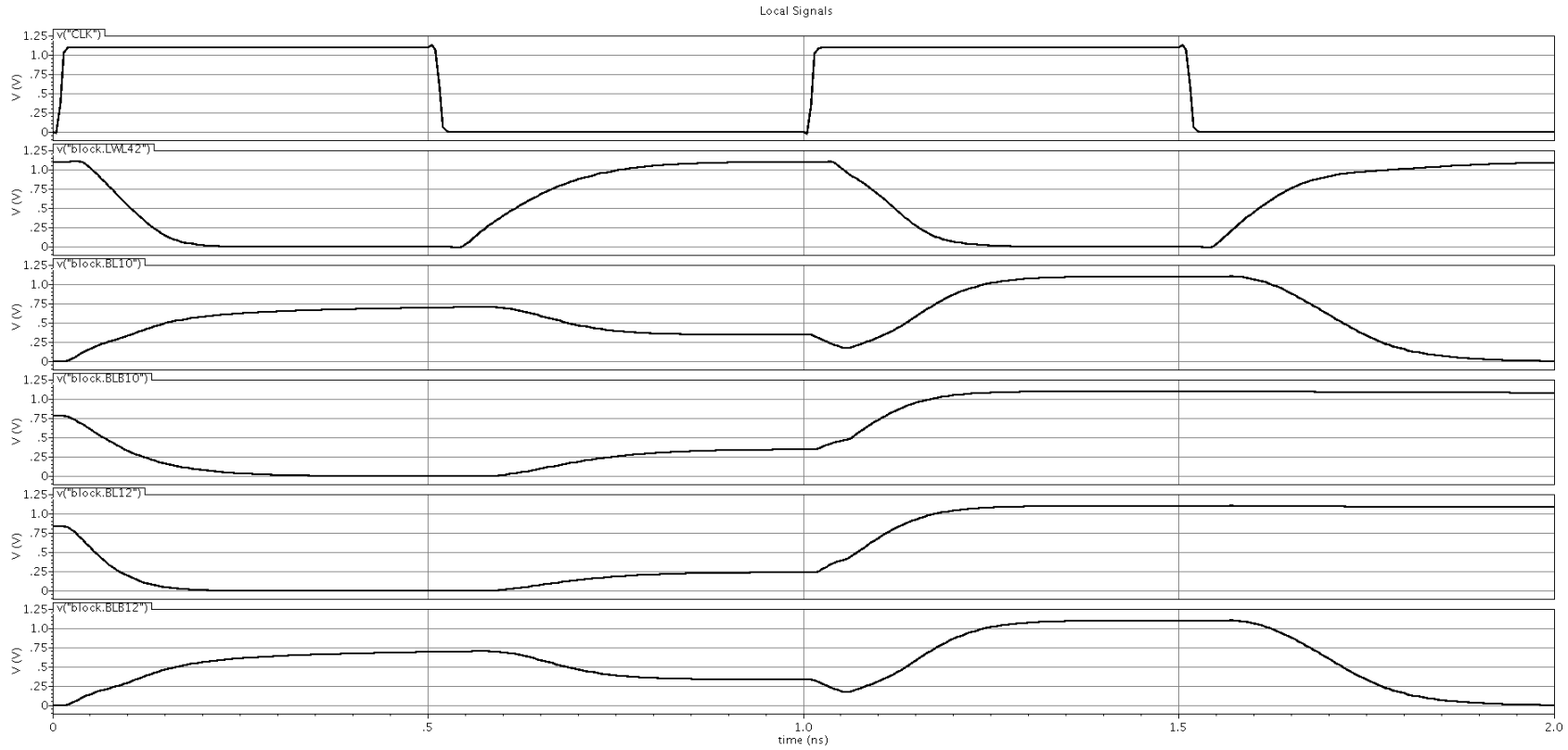

Figure 6 WL, BL and BLB during TT Write and Read Operations

Figure 7 shows the input data IN <13:10> and the output data OUT<13:10> during the Typical-Typical write and read operations. Bit 13 to 10 are randomly selected bits to demonstrate the functionality of the cache. The output data OUT are valid during the last half clock period when the control signal to the sense amplifiers SE is enabled as shown in Figure 5. On the 2nd clock cycle, the output successfully reads the input. Only for the TT case, we added an inverter at the end of the output in order to receive the correct values.

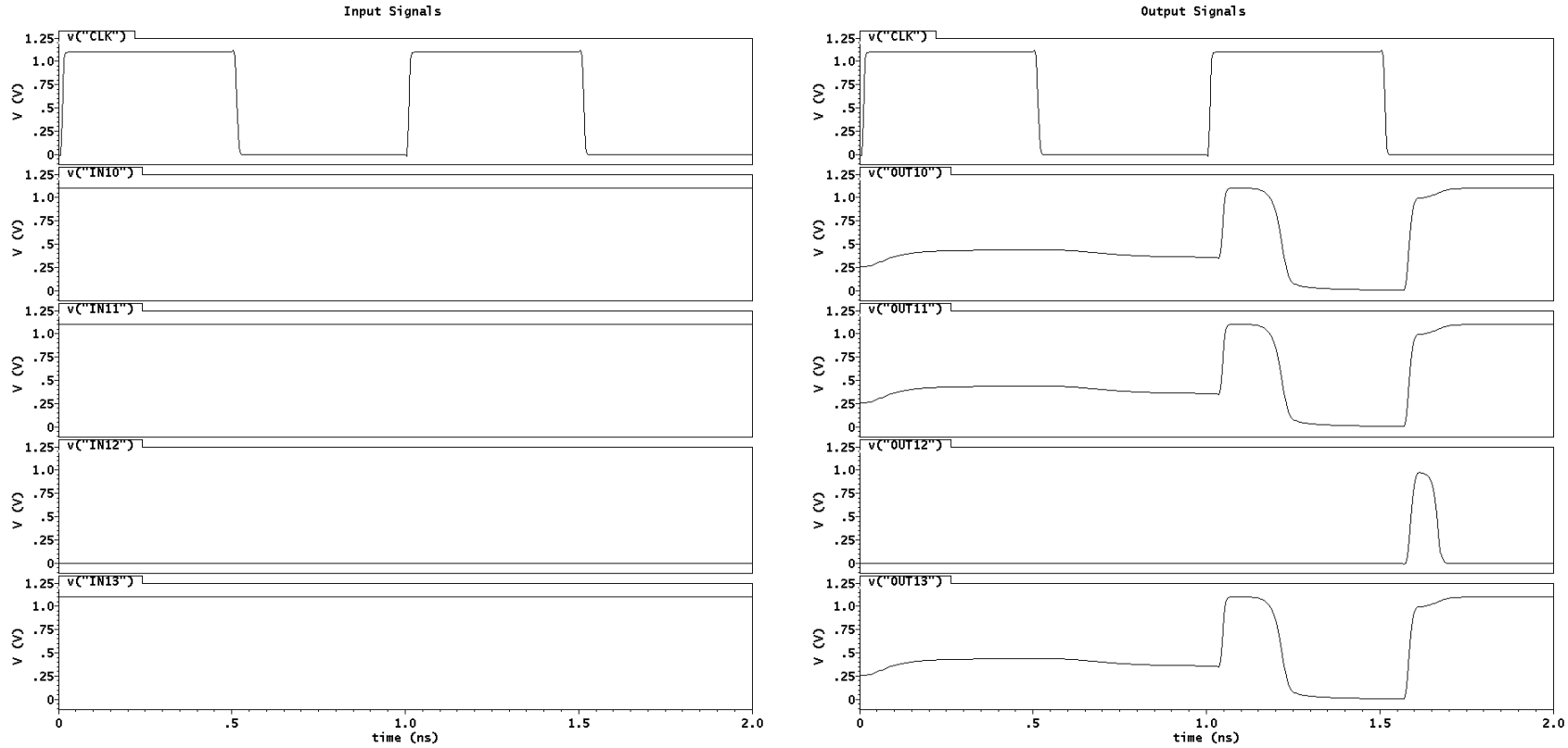

Figure 7 IN<13:10> and OUT<13:10> during TT Write and Read Operations

Figure 8 shows the local wordline WL42, BL/BLB10 and BL/BLB12 of the cache during write and read operations at the Fast-Fast corner.

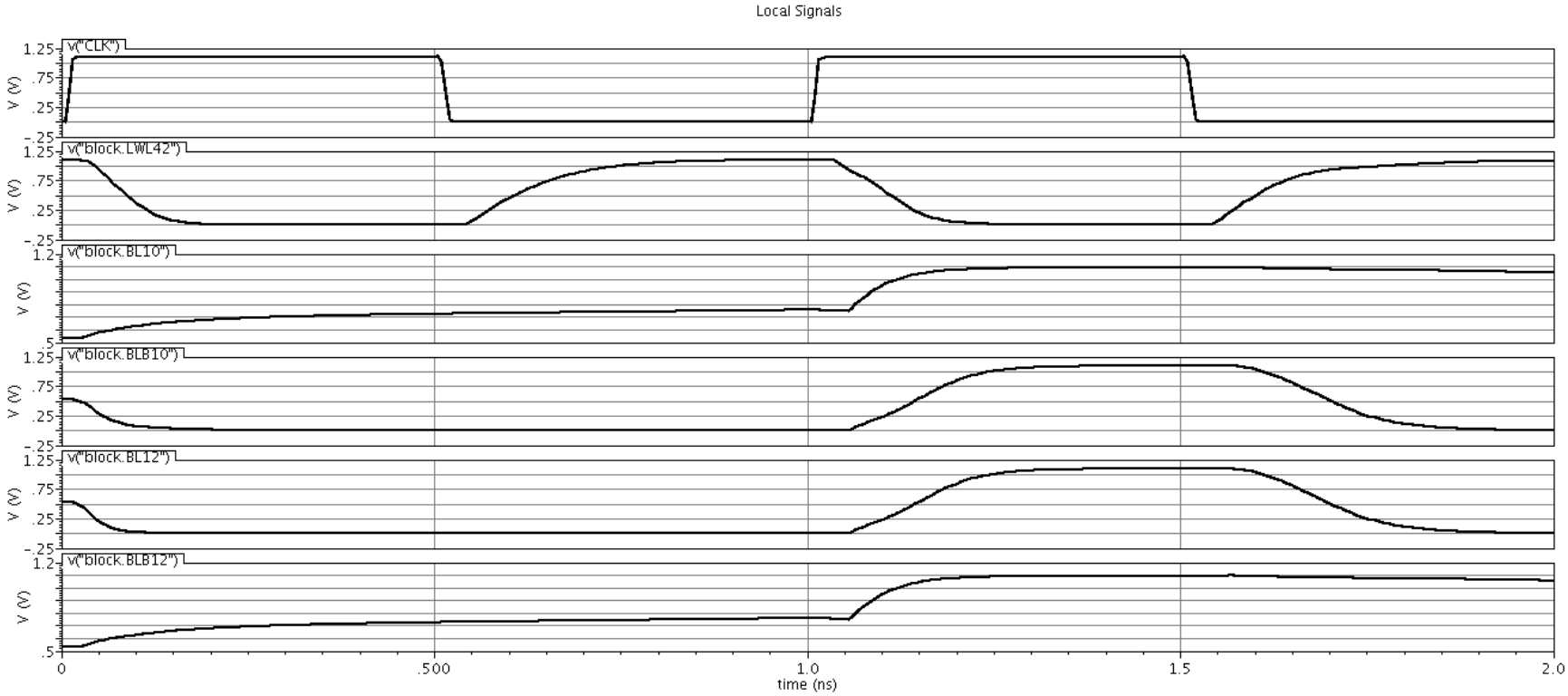

Figure 8 WL, BL and BLB during FF Write and Read Operations

Figure 9 shows the input data IN <13:10> and the output data OUT<13:10> during the write and read operations at the Fast-Fast corner.

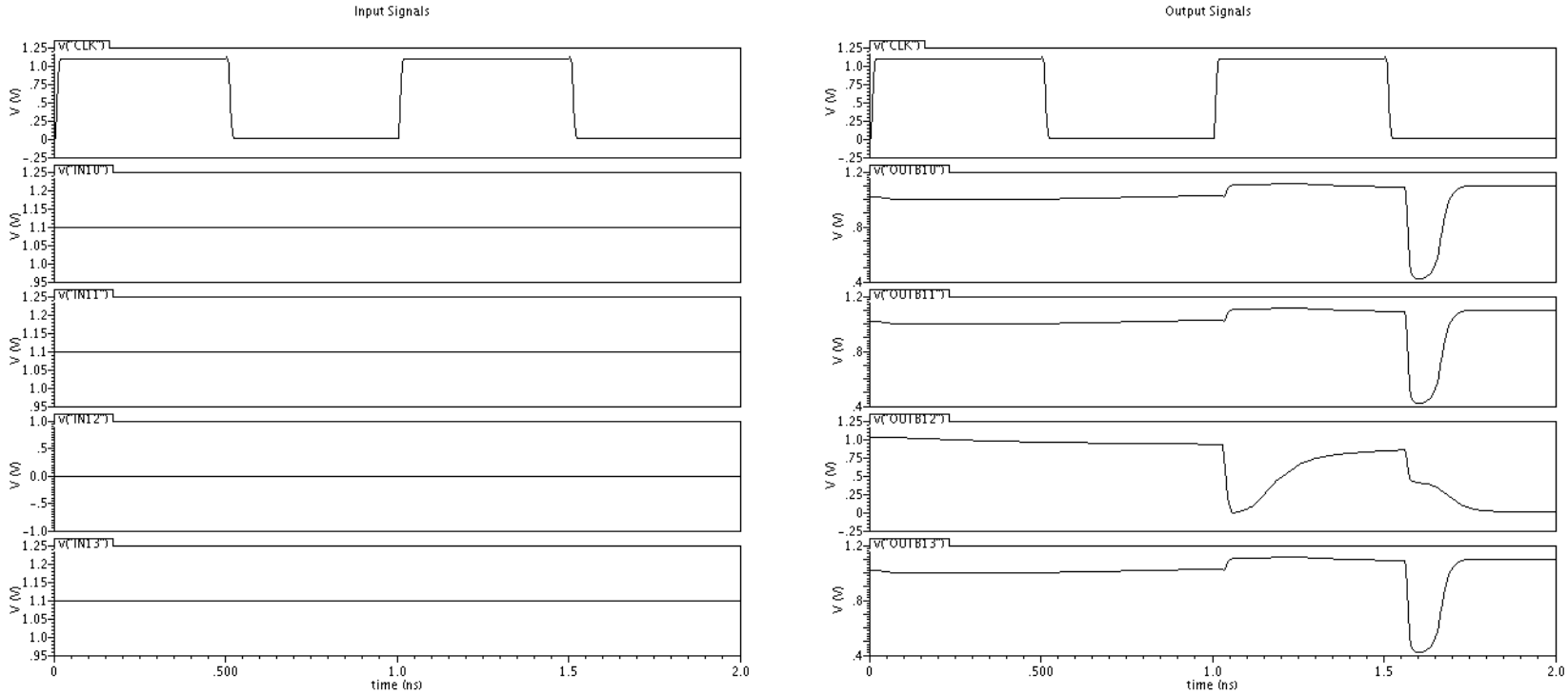

Figure 9 IN<13:10> and OUT<13:10> during FF Write and Read Operations

Progress and Remaining tasks

So far, we have decided on the basic architecture of our SRAM, which is shown above in figure 1. We have built decoders for the address lines of the rows, columns, and blocks of our SRAM. We have constructed a sense amplifier, for reading from the bit lines. We have designed and sized a 6T SRAM bit cell, and used that bit cell to construct a 64x64 bit array of bit cells (which we define as a block). We have also created a layout for that 6T bit cell. We have connected all of our components together in a netlist to create a 4 kbit SRAM, and we have simulated both write and read operations to verify functionality. We have also been regularly updating our wiki page with our progress.

Before the proposal is due next week, we must research the designs of previous groups so that we can improve upon them. We must also research, and then decide on what novelty features we plan to implement in our design. We will then have to meet with Professor Calhoun to get our ideas approved before submitting our proposal.

After the proposal is submitted, we must extend our 4 kbit SRAM to 64 kbits. We must then optimize our design for the metric specified in the project assignment, taking into account the process corners. We will also have to create a layout of our design.